

WHAT IS CLAIMED IS:

- 1 1. A computer implemented method for automatically
2 nullifying variables in a middleware computer program,
3 said method comprising:
4 reading one or more variables included in one or more
5 activation records included in the computer program;
6 identifying a program statement in the program where
7 the variable is last used; and
8 inserting a nullification statement after the
9 identified program statement, the nullification
10 statement adapted to nullify the identified last-used
11 variable.
- 1 2. The method of claim 1 wherein the reading,
2 identifying, and inserting are each performed by a
3 compiler.
- 1 3. The method of claim 1 further comprising:
2 writing the activation records, program statement, and
3 nullification statement to a resulting code file.
- 1 4. The method of claim 1 wherein at least one of the
2 variables reference an object stored in a garbage collected
3 memory heap.
- 1 5. The method of claim 1 wherein the activation records
2 include one or more local variable definitions.
- 1 6. The method of claim 1 wherein the activation records
2 include one or more argument parameters.

1 7. The method of claim 1 wherein the objects are stored
2 in a garbage collected heap stored in a computer memory,
3 the method further comprising:
4 executing a garbage collection program;
5 identifying, by the garbage collection program, one of
6 the objects that was previously referenced by one of
7 the variables included in the nullification statement;
8 and
9 reclaiming the memory occupied by the identified
10 object.

1 8. The method of claim 1 further comprising:
2 executing a compiler to perform the reading,
3 identifying and inserting;
4 writing a plurality of program statements, including
5 the program statement, to a resulting code file;
6 writing the nullification statement to the resulting
7 code file in a position subsequent to the identified
8 program statement.

1 9. The method of claim 8 further comprising:
2 identifying one or more statements from the plurality
3 of statements where one or more other objects are last
4 used; and
5 writing nullification statements for each of the other
6 objects following the identified statement

7 corresponding to the object's last use to the
8 resulting code file.

1 10. An information handling system comprising:

2 one or more processors;

3 a memory accessible by the processors;

4 a middleware software application that runs on the
5 operating system, the middleware application including
6 a garbage-collected heap; and

7 a nullification tool for nullifying program
8 references, the nullification tool comprising steps
9 effective to:

10 read one or more variables included in one or
11 more activation records included in the computer
12 program;

13 identify a program statement in the program where
14 the variable is last used; and

15 insert a nullification statement after the
16 identified program statement, the nullification
17 statement adapted to nullify the identified last-
18 used variable..

1 11. The information handling system of claim 10 wherein
2 the nullification tool is a compiler.

1 12. The information handling system of claim 10, wherein
2 the nullification tool is further effective to:

3 write the activation records, program statement, and
4 nullification statement to a resulting code file.

1 13. The information handling system of claim 10 wherein at
2 least one of the variables reference an object stored in a
3 garbage collected memory heap.

1 14. The information handling system of claim 10 further
2 comprising a garbage collected heap stored in the memory,
3 wherein the steps are further effective to:

4 execute, by the processors a garbage collection
5 program;

6 identify, by the garbage collection program, one of
7 the objects that was previously referenced by one of
8 the variables included in the nullification statement;
9 and

10 reclaim the memory occupied by the identified object.

1 15. A computer program product stored in a computer
2 operable media for automatically nullifying variables in a
3 middleware computer program, said computer program product
4 comprising:

5 means for reading one or more variables included in
6 one or more activation records included in the
7 computer program;

8 means for identifying a program statement in the
9 program where the variable is last used; and

10 means for inserting a nullification statement after
11 the identified program statement, the nullification

12 statement adapted to nullify the identified last-used
13 variable.

1 16. The computer program product of claim 15 wherein the
2 means for reading, means for identifying, and means for
3 inserting are each performed by a compiler.

1 17. The computer program product of claim 15 further
2 comprising:

3 means for writing the activation records, program
4 statement, and nullification statement to a resulting
5 code file.

1 18. The computer program product of claim 15 wherein at
2 least one of the variables reference an object stored in a
3 garbage collected memory heap.

1 19. The computer program product of claim 15 wherein the
2 activation records include one or more local variable
3 definitions.

1 20. The computer program product of claim 15 wherein the
2 activation records include one or more argument parameters.

1 21. The computer program product of claim 15 wherein the
2 objects are stored in a garbage collected heap stored in a
3 computer memory, the method further comprising:

4 means for executing a garbage collection program;

5 means for identifying, by the garbage collection
6 program, one of the objects that was previously
7 referenced by one of the variables included in the
8 nullification statement; and

9 means for reclaiming the memory occupied by the
10 identified object.

1 22. The computer program product of claim 15 further
2 comprising:

3 means for executing a compiler to perform the reading,
4 identifying and inserting;

5 means for writing a plurality of program statements,
6 including the program statement, to a resulting code
7 file;

8 means for writing the nullification statement to the
9 resulting code file in a position subsequent to the
10 identified program statement.

1 23. The computer program product of claim 15 further
2 comprising:

3 means for identifying one or more statements from the
4 plurality of statements where one or more other
5 objects are last used; and

6 means for writing nullification statements for each of
7 the other objects following the identified statement
8 corresponding to the object's last use to the
9 resulting code file.

1 24. A method for automatically nullifying variables in a
2 middleware computer program, said method comprising:

3 reading one or more variables included in one or more
4 activation records included in the computer program;

5 identifying a program statement in the program where
6 the variable is last used;

7 inserting a nullification statement after the
8 identified program statement, the nullification
9 statement adapted to nullify the identified last-used
10 variable;

11 writing a plurality of program statements, including
12 the identified program statement, to a resulting code
13 file; and

14 writing the nullification statement to the resulting
15 code file in a position subsequent to the identified
16 program statement.

1 25. An information handling system comprising:

2 one or more processors;

3 a memory accessible by the processors;

4 a middleware software application that runs on the
5 operating system, the middleware application including
6 a garbage-collected heap; and

7 a nullification tool for nullifying program
8 references, the nullification tool comprising steps
9 effective to:

10 read one or more variables included in one or
11 more activation records included in the computer
12 program;

13 identify a program statement in the program where
14 one of the variables is last used;

15 insert a nullification statement after the
16 identified program statement, the nullification
17 statement adapted to nullify the identified last-
18 used variable;

19 write a plurality of program statements,
20 including the identified program statement, to a
21 resulting code file; and

22 write the nullification statement to the
23 resulting code file in a position subsequent to
24 the identified program statement.

1 26. A computer program product stored in a computer
2 operable media for automatically nullifying objects in a
3 middleware computer program, said computer program product
4 comprising:

5 means for reading one or more variables included in
6 one or more activation records included in the
7 computer program;

8 means for identifying a program statement in the
9 program where the variable is last used;

10 means for inserting a nullification statement after
11 the identified program statement, the nullification
12 statement adapted to nullify the identified last-used
13 variable;

14 means for writing a plurality of program statements,
15 including the identified program statement, to a
16 resulting code file; and

17 means for writing the nullification statement to the
18 resulting code file in a position subsequent to the
19 identified program statement.